Debreceni Egyetem
Informatikai Kar

ECC alapú elektronikus szavazási séma

Varga Péter

2008

Témavezető: Dr. Pethő Attila
egyetemi tanár

Ezúton szeretnék köszönetet mondani témavezetőmnek, Dr. Pethő Attila tanár úrnak a szakmai irányításáért, és hogy lehetőséget biztosított dolgozatom elkészítéséhez. Köszönöm a rengeteg türelmet, és a hasznos tanácsokat, amik nélkül ez a dolgozat nem jöhetett volna létre.

Tartalomjegyzék

1. fejezet

Bevezetés

A kriptográfia ógörög eredetű szó, jelentése: titkosírás. Ez a tudomány azóta foglalkoztatja az embert, amióta a tulajdon, mint fogalom, a birtoklás részévé vált az emberiségnek, azonban tényleges kutatása csak néhány évtizeddel ezelőtt kezdődött meg. Régebben a nyelvtudomány témakörébe sorolták, azonban mára már erősen matematikai és informatikai jelleget kapott. Rejtjelezéssel, titkosírással, kódolással foglalkozik, melyek előállítása mellett egy másik fő cél azok megfejtése is. Az elliptikus görbéket, mint algebrai és geometriai elemet ugyan az elmúlt 150 évben behatóan tanulmányozták, azonban kriptográfiai szempontból még csak 20 éve vizsgálják őket.
Jelen dolgozat célja az elliptikus görbék által alkotott kriptorendszer (röviden: ECC) alkalmazása elektronikus szavazást megvalósító protokollok esetén.

A második fejezetben alapfogalmakkal ismerkedhetünk meg, melyek segítséget nyújtanak a dolgozat lényegének megértésében. A fő cél az elliptikus görbe, mint fogalom megismerése, megértése.
A harmadik fejezet átvezetőnek tekinthető a második fejezetben megismert elliptikus görbék, és ezek kriptográfiai alkalmazása között. Ez a fejezet véges testek felett értelmezett elliptikus görbékkel foglalkozik, bevezeti a Hasse-tételt, valamint a Diszkrét Elliptikus Logaritmus Problémát, amely lehetővé teszi az ECC, mint kriptorendszer alkalmazását.
A negyedik fejezetből megtudhatjuk, hogyan is működik ez a kriptorendszer, milyen paraméterei vannak az eljárásnak.
Az ötödik fejezet úgynevezett átlátszó bizonyításokkal foglalkozik, amelyek segítségével úgy bizonyíthatjuk egy titkos adat ismeretét, hogy eközben semmilyen információt nem adunk ki az adatról. Ilyen eljárások elengedhetetlenek egy elektronikus szavazási rendszer implementálásához.

A hatodik fejezet a szavazási eljárás alapfogalmait tárgyalja, valamint azt, hogy milyen követelményeknek kell teljesülni egy ilyen rendszer működése során.
A hetedik, nyolcadik és kilencedik fejezet elliptikus görbe alapú elektronikus szavazási sémákat tárgyal. A hetedik fejezetben egy egyszerűbb szavazási sémát ismerhetünk meg, melynek segítségével megérthetjük ezen eljárások menetét.
A nyolcadik és kilencedik fejezetekben ennek a sémának az általánosításait olvashatjuk.
A tizedik fejezet az implementációval foglalkozik, leírja, hogyan sikerült a kilencedik fejezetben tárgyalt sémát megvalósítani.

2. fejezet

Alapfogalmak

Ebben a fejezetben megismerkedünk azokkal az alapfogalmakkal, amelyek ismerete elengedhetetlen a véges testek feletti elliptikus görbéken alapuló titkosítási eljárás megértéséhez.

2.1. Definíció (Hosszú Weierstraß normálforma)[1]

Legyen

$$E\colon\ Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6$$

egy $\mathbb{K}$ test felett értelmezett egyenlőség, ahol $a_1, a_2, a_3, a_4, a_6, X, Y \in \mathbb{K}$. Ekkor azt mondjuk, hogy E hosszú Weierstraß normálformában van.

2.2. Definíció (Tate értékek)[1]

Legyen E egy a_1, a_2, a_3, a_4, a_6 együtthatókkal megadott hosszú Weierstraß normálformában lévő egyenlőség. Ekkor E Tate értékeinek nevezzük a következőket:

$$b_2 := a_1^2 + 4a_2,$$

$$b_4 := 2a_4 + a_1a_3,$$

$$b_6 := a_3^2 + 4a_6,$$

$$b_8 := a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2,$$

$$c_4 := b_2^2 - 24b_4,$$

$$c_6 := -b_2^3 + 36b_2b_4 - 21b_6.$$

A diszkrimináns:

$$\Delta := -b_2^2 b_8 - 8b_4^3 - 27b_6^3 + 9b_2 b_4 b_6.$$

és a j-invariáns:

$$j := \frac{c_4^3}{\Delta}.$$

2.3. Definíció (Elliptikus görbe)[1]

Legyenek $a_1, a_2, a_3, a_4, a_6, X, Y \in \mathbb{K}$. Azon $(x, y) \in \mathbb{K}$ pontok halmazának, melyekre

$$Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6$$

teljesül és az $\{\mathcal{O}\}$ halmaznak az unióját hosszú Weierstraß normálformában megadott elliptikus görbének nevezzük.

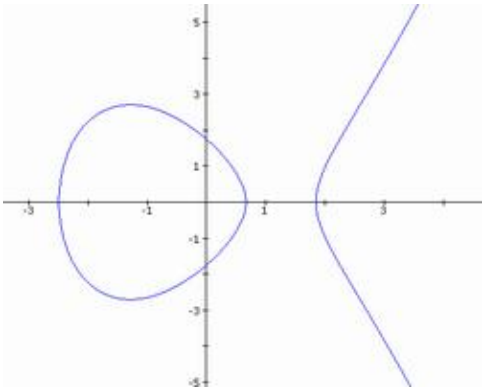
2.4. Definíció (Rövid Weierstraß normálforma)[1]

Legyen

$$Y^2 = X^3 + a_4X + a_6$$

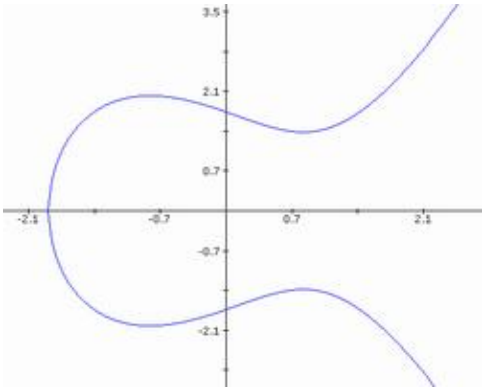
egy $\mathbb{K}$ test felett értelmezett egyenlet, ahol $a_4, a_6, X, Y \in \mathbb{K}$. Ekkor azt mondjuk, hogy ez az egyenlet rövid Weierstraß normálformában van.

Hosszú Weierstraß normálformában megadott elliptikus görbékre a következő osztályozást vezetjük be:
$\diamond$ Az elliptikus görbe reguláris akkor és csak akkor, ha $\Delta \neq 0$. Egyébként szinguláris pontosan egy szinguláris ponttal (a pontban a görbe mindkét változó szerinti deriváltja 0).
Például a következő két görbe reguláris:



(a)

2.1. ábra. $y^2 = x^3 - 5x + 3$



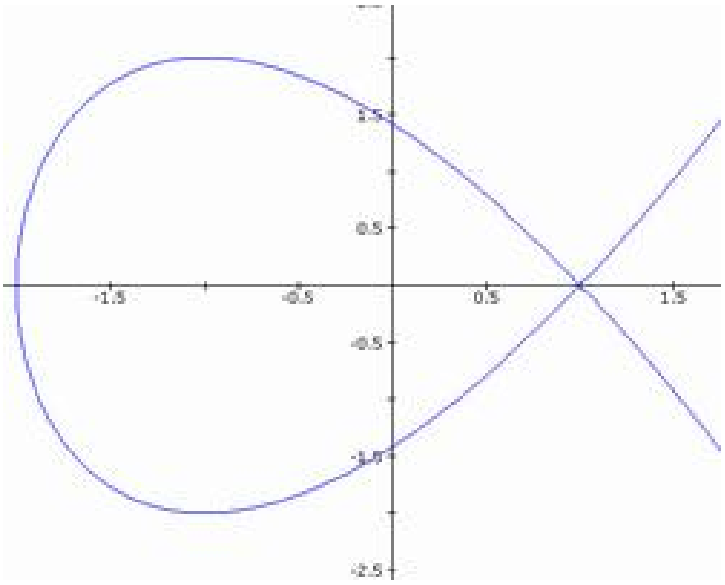
(a)

2.2. ábra. $y^2 = x^3 - 2x + 3$

Ezen görbék diszkriminánsai a valós számhalmaz felett értelmezve $\Delta = -257$, illetve $\Delta = 211$.

$\diamond$ Az elliptikus görbének csomópontja (node) van, akkor és csak akkor, ha $\Delta = 0$ és $c_4 \neq 0$.

Például: Ezen görbére: $D = 0$, $c_4 = 144$ és csomópontja (szinguláris pon-



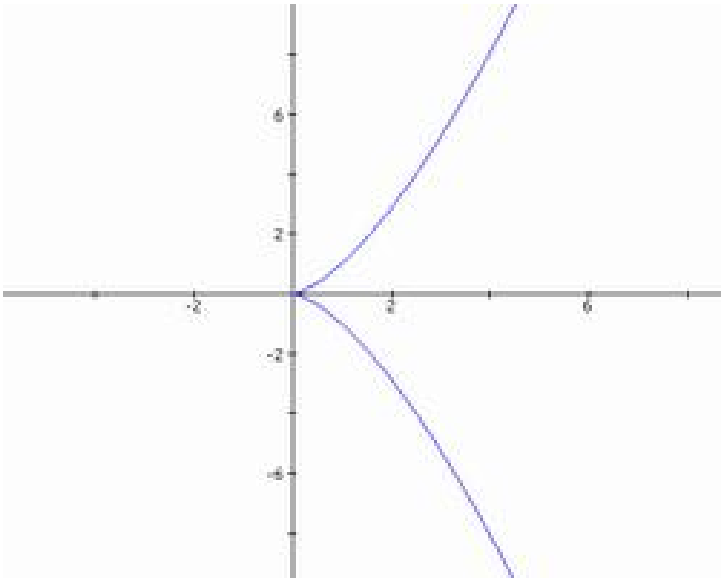
(a)

2.3. ábra. $y^2 = x^3 - 3x + 2$

tja): $(x, y) = (1, 0)$.

$\diamond$ Az elliptikus görbének fordulópontja (cusp) van, akkor és csak akkor, ha $\Delta = 0$ és $c_4 = 0$.

Például:



(a)

2.4. ábra. $y^2 = x^3$

Ezen görbére: $D = 0$, $c_4 = 0$ és fordulópontja (szinguláris pontja): $(x, y) = (0, 0)$.

2.5. Definíció (Görbék ekvivalenciája)[1]

Azt mondjuk, hogy az E és F ugyanazon test felett értelmezett elliptikus görbék egymással ekvivalensek, ha egymásba átvihetők véges sok mértéktartó lineáris transzformáció alkalmazásával. Ez egy ekvivalencia reláció az ugyanazon test felett értelmezett görbékre, ami osztályozást hoz létre a test felett értelmezett elliptikus görbék halmazán.

Tétel[1]

Ha a $\mathbb{K}$ test karakterisztikája > 3, akkor bármely hosszú Weierstraß normálformában lévő egyenlettel megadott elliptikus görbéhez megadható egy vele ekvivalens rövid Weierstraß normálformában lévő egyenlettel megadott elliptikus görbe.

Bizonyítás:
A bizonyítás konstruktív. Induljunk ki a hosszú Weierstraß normálformájú egyenletek általános alakjából:

$$E\colon\ Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6$$

Új változók bevezetésével kapjuk a következőket:

$$Y' = Y + \frac{a_1}{2}X + \frac{a_3}{2}$$

$$X' = X + \frac{4a_2 + a_1^2}{12}$$

Ezzel az egyenlet a következő alakba jut:

$$Y'^2 = X'^3 + (a_4 + \frac{a_1a_3}{4} - \frac{(4a_2 + a_1^2)^2}{48})X' + (a_6 + \frac{a_3^2}{4} - \frac{(4a_2 + a_1^2)^3}{1728})$$

2.6. Definíció (Műveletek elliptikus görbéken)[1]

Az E

$$Y^2 = X^3 + a_4X + a_6$$

rövid Weierstraß normálformában lévő egyenlettel megadott elliptikus görbén műveleteket lehet definiálni a következőképpen:
Pontok összeadása:
Legyenek

$$P = (x_1, y_1), Q = (x_2, y_2), R = (x_3, y_3) \in E$$

az elliptikus görbe pontjai. Azt mondjuk, hogy $P+Q = R$, ha a következők közül az egyik teljesül:
$\diamond$ $x_1 = x_2$ és $y_1 = -y_2$ és $R = \mathcal{O}$
$\diamond$ $P = \mathcal{O}$ és $Q = R$ vagy $Q = \mathcal{O}$ és $P = R$
$\diamond$ $x_3 = \lambda^2 - x_1 - x_2$ és $y_3 = \lambda(x_1 - x_3) - y_1$ és

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{, ha } P \neq Q \\ \frac{3x_1^2 + a_4}{2y_1} & \text{, ha } P = Q \end{cases}$$

Pontok skalárszorosa:
A P pont $\gamma \in Z^+$ skalárszorosán azt a Q pontot értjük, melyre

$$Q = \sum_{i=1}^{\gamma} P$$

Azaz a P pontot γ-szor önmagához adva Q-t kapjuk.
Jele: $[\gamma]P$

Tétel[1]

Egy E elliptikus görbe az előző definícióban megadott műveletre Abel-csoportot alkot.

3. fejezet

Véges test felett értelmezett elliptikus görbék

Eddig általánosan beszéltünk az elliptikus görbékről. Most áttekintjük azt, mit mondhatunk az elliptikus görbékről, ha speciálisan csak véges testek felett vizsgáljuk őket.

Legyen $\mathbb{K}$ véges test, és legyen E $\mathbb{K}$ felett értelmezett elliptikus görbe. Ekkor az E elliptikus görbe véges Abel-csoportot alkot, mely vagy ciklikus, vagy két ciklikus csoport direkt szorzata.

Ekkor

3.1. Hasse-tétel[1]

Legyen E elliptikus görbe $\mathbb{K}_q$ véges test felett.
Ekkor

$$|\#E - (q+1)| \leq 2\sqrt{q},$$

ahol $\#E$ az E elemeinek számát jelenti.

3.2. Definíció (Diszkrét Elliptikus Logaritmus Probléma)[1]

Legyen G az E elliptikus görbe által alkotott ciklikus csoport, rendje $m = ord(G)$. Legyen adott $g, h \in G, h = [l]g, 0 \leq l \leq m$. Diszkrét Elliptikus Logaritmus Problémának nevezzük azt, amikor g, h, m ismeretében meg kell határoznunk az l értéket.

Ma nem ismert olyan módszer, mellyel mindez polinomiális időben kiszámolható, a leggyorsabb ismert algoritmus bonyolultsága szubexponenciális.

3.3. A Diszkrét Elliptikus Logaritmus Probléma megoldása[1]

A Diszkrét Logaritmus Probléma megoldható abban az esetben, ha tudjuk, hogy a végeredmény egy bizonyos tartományba esik, amely egy viszonylag kevés elemszámú halmaz. Legyenek ezek a halmazbeli értékek: $0, 1, ..., p-1$.

3.3.1. Próbálgatás módszere

Kiszámoljuk $[i]g$-t minden $i = 0, 1, ..., p-1$-re, amíg $h = [i]g$ nem teljesül, ahol p szigorúan kisebb, mint m. Maximum p esetet kell végigpróbálni.

3.3.2. D. Shank "baby-step giant-step" algoritmusa

Legyen adott egy $o < p$ szám. Számítsuk ki, és tároljuk

$$BS[j] = h + [-j]g = [l - j]g$$

$0 \leq j < o$-re, valamint számítsuk ki a

$$GS[k] = [k \cdot o]g$$

pontokat $k = 0 \leq k < [\frac{p}{o}]$-ra és hasonlítsuk össze a két ponthalmazt. Ha van olyan j és k, melyekre $BS[j] = GS[k]$, az azt jelenti, hogy

$$[l - j]g = [k \cdot o]g \Rightarrow [l - j - ko]g = \{\mathcal{O}\} \Rightarrow l = j + k \cdot o.$$

Így megkaptuk i értékét. A legköltségesebb művelet az algoritmusban a pontösszeadás, ezeknek a száma:

$$o + [\frac{p}{o}] \approx \frac{o^2 + p}{o}.$$

Ez o egy függvénye, melyben p paraméter. Mivel $p > 0$, ezért ez a függvény egy hiperbola, melynek csak a pozitív ágát vizsgáljuk, mert $o > 0$. Ennek az ágnak minimuma lesz, ha a derivált zérus értéket vesz fel.

$$f(o) = \frac{o^2 + p}{o},$$

$$f'(o) = \frac{o^2 - p}{o^2}.$$

Mivel a derivált függvény $o = \sqrt{p}$ esetben lesz nulla, így célszerű o-t $\sqrt{p}$-nek választani.

4. fejezet

ECC - Elliptikus görbéken alapuló titkosítási séma[1]

A bevezető tételek és definíciók után nézzük meg, hogyan használhatóak az elliptikus görbék elektronikus üzenetek titkosítására.

A Diszkrét Elliptikus Logaritmus Problémát kihasználva felépíthető egy elliptikus görbéket alkalmazó titkosítási algoritmus. Legyen A (Alice) az a személy, aki titkos csatornán szeretne információt megosztani B-vel (Bob). E (Eve) a "támadó", aki ezt az üzenetet szeretné feltörni.
Legyen az átküldendő üzenet m darab karakter.
B választ egy $E(\mathbb{K}_q)$ véges elliptikus görbét ($|E(\mathbb{K}_q)| = r$) , ennek egy $G \in E(\mathbb{K}_q)$ generátorelemét, valamint egy $0 \leq n < r$ titkos kulcsot, és nyilvánosságra hozza $E(\mathbb{K}_q)$-t, G-t és $[n]G$-t.
A leképezi az átküldendő üzenetet a következőképpen:

Az üzenetet részekre bontja szét ($2 \cdot [\log q]$ bitekre) és minden egyes részhez meghatározza a hozzá tartozó $P_i \in \mathbb{K}_q^2$, $0 \leq i \leq m-1$ pontot, úgy, hogy az üzenetet két részre vágja, és az üzenet első fele lesz a pont x koordinátája, az üzenet második fele pedig az y. Generál egy $k \in \mathbb{Z}$ véletlen értéket, majd létrehozza a

$$([k]G, P_i + [k]([n]G))$$

párokat, ahol a + jel itt nem az elliptikus görbéken történő pontösszeadást jelenti, hiszen P_i nem feltétlenül van rajta a görbén. Ez az összeadás koordinátánkénti összeadást jelöl. Az elkészült üzenetet átküldi B-nek.
B tudja olvasni az üzenetet, mivel ismeri n-t, így ki tudja számolni $[nk]G$-t, ezt koordinátánként kivonva az üzenet második feléből megkapja P_i-t.

E nem tudja olvasni, mivel sem n-t, sem k-t nem ismeri, így nem tudja $[nk]G$-t meghatározni.

5. fejezet

Zero-knowledge proofs - Átlátszó bizonyítások

Ebben a fejezetben két olyan bizonyítási eljárást ismertetünk, melyek feladata kettős:

- a vizsgáló személy megbizonyosodjon arról, hogy egy másik személy ténylegesen olyan adatot titkosított, amely egy bizonyos kritériumnak megfelel.

- a bizonyítónak ne kelljen semmilyen tényleges információt elárulnia a titkosított adatról.

5.1. Logikai kifejezés bizonyítása

Ebben a részben egy olyan protokollt tárgyalunk, amely azt bizonyítja, hogy egy L összetett kvantormentes logikai kifejezés értéke igaz, oly módon, hogy a bizonyító ne áruljon el semmit a predikátumok értékéről.
L-ben szereplő minden egyes predikátum két diszkrét elliptikus logaritmikus kifejezés egyenlőségét vizsgálja. Mind a bizonyító, mind a vizsgáló ismeri a diszkrét logaritmikus kifejezéseket, viszont a vizsgáló nem ismeri ezek értékét, ennek kiszámolása diszkrét elliptikus logaritmus probléma, ezért nem tudja, hogy egyenlőek e a kifejezések.
Hozzuk L-et konjunktív normálformára. Ekkor L értéke akkor lesz igaz, ha

minden egyes elemi diszjunkció értéke igaz. Legyen egy ilyen diszjunkció:

$$\bigvee_{i=1}^{n} \log_{g_i} h_i = \log_{G_i} H_i.$$

A bizonyító megmutatja, hogy ő ismeri a diszkrét elliptikus logaritmusok eredményét, és hogy a fenti kifejezés igaz, úgy, hogy a vizsgáló ne jusson semmilyen információhoz, ami segíthetné a diszkrét elliptikus logaritmusok kiszámolásában. A következő algoritmust használja:
Legyenek
t = a diszjunkcióban azon predikátumnak a sorszáma, amely igaz értéket vesz fel
v = a t. predikátumban szereplő logaritmikus kifejezések értéke (a predikátum igaz volta miatt ez a két logaritmus megegyezik)
A bizonyító legenerálja a d és r random vektorokat, és kiszámolja a következőket:

$$d \in Z_p^n,$$

$$r \in Z_p^n,$$

$a, b \in E^n$ n dimenziós pontvektorok, ahol

$$a_i = [d_i]h_i + [r_i]g_i,$$

$$b_i = [d_i]H_i + [r_i]G_i,$$

$$w = v \cdot d_t + r_t.$$

Elküldi az a és b vektorokat a vizsgálónak. A vizsgáló generál egy $c \in Z_p$ számot, majd ezt küldi vissza. A bizonyító felülírja a d_t és az r_t értékeket a következőkkel:

$$d_t = c - \sum_{j \neq t} d_j,$$

$$r_t = w - v \cdot d_t,$$

majd elküldi a d és r vektorokat a vizsgálónak. A vizsgáló megnézi, hogy

$$c = \sum_{j=1}^{n} d_j,$$

$$a_i = [d_i]h_i + [r_i]g_i,$$

$$b_i = [d_i]H_i + [r_i]G_i,$$

egyenlőségek teljesülnek e. Ha igen, akkor a bizonyító bizonyította állítását, mivel $j \neq t$-re d_j és r_j értékei változatlanok, így a vizsgáló által kiszámított értékek megegyeznek a_j-vel és b_j-vel. Ha $j = t$, akkor d'_t és r'_t jelölje d_t és r_t új értékét. A vizsgáló által számított érték:

$$a'_t = [d'_t]h_t + [r'_t]g_t = [v \cdot d'_t]g_t + [w - v \cdot d'_t]g_t = [w]g_t$$

$$= [v \cdot d_t + r_t]g_t = [d_t][v]g_t + [r_t]g_t = [d_t]h_t + [r_t]g_t = g_t.$$

b_t-re az állítás hasonlóképpen bizonyítható.

5.2. Diszkrét elliptikus logaritmus ismeretének bizonyítása

Adott egy $L = \log_g G = \log_h H$ diszkrét elliptikus logaritmikus egyenlet. Mind a bizonyító, mind a vizsgáló ismeri a g, G, h, H pontokat, de L-et csak a bizonyító ismeri.
Ezt az ismeretet szeretné bizonyítani, oly módon, hogy a vizsgáló ne jusson semmilyen információhoz, ami segíthetné a diszkrét logaritmus kiszámolásában. A következő algoritmust használja:

A bizonyító generál egy véletlen számot $\omega \in Z_p$, kiszámolja $(a, b) = ([\omega]g, [\omega]h)$ pontokat, kiszámolja $c = Hash(a||b||G||H)$-t, ahol $Hash$ egy biztonságos hash-függvény. Meghatározza az $r = \omega + L \cdot c$ értéket, és c-t és r-t küldi el a vizsgálónak. A vizsgáló kiszámolja $H([r]g - [c]G||[r]h - [c]H||G||H)$-t, és megnézi, hogy ez egyenlő e c-vel. Ha igen, akkor a vizsgáló biztos lehet benne, hogy a bizonyító ismeri a logaritmus eredményét.

6. fejezet

Szavazási eljárás és a vele szemben támasztott követelmények[2]

Tekintsük át, milyen alapfogalmai és kritériumai vannak általában egy szavazási eljárásnak.

6.1. Szavazás

Szavazásnak nevezzük azt a folyamatot, amikor egy közösség minden egyes tagja előre megadott kérdésekről, témákról dönt.

6.2. Szavazási témák

Azon kérdéskörök, amelyekben a közösség tagjainak döntést kell hoznia. Minden egyes témához előre megadott válaszlehetőségek vannak megfogalmazva.

6.3. Szereplők

6.3.1. Szavazók

Azok, akik a szavazás során jogosultak szavazni.

6.3.2. Bírák (Bizottsági tagok)

Azon személyek, akik a szavazás tisztaságát biztosítják.

6.4. Szavazat

Egy szavazó egy témában adott válasza. Eleme kell legyen az előre megadott válaszlehetőségek halmazának. Minden szavazó minden kérdésben csak egy választ adhat.

6.5. Kiértékelés

Az a folyamat, amikor a bírák a szavazatokat összeszámlálják, és eldöntik, hogy az egyes témákban mi az, amit a legtöbben választottak, ezt nevezzük az adott témában történő szavazás eredményének.

6.6. Szavazási eljárás

A szavazás, és a kiértékelés együttes folyamatát szavazási eljárásnak hívjuk.

6.7. Követelmények

6.7.1. Választhatóság

Csak azon személyek szavazhassanak, akik jogosultak rá, valamint ők is minden témában csak egyszer szavazhassanak.

6.7.2. Anonimitás

Egyik leadott szavazatról se lehessen visszakövetni a szavazó személyét.

6.7.3. Titkosság

A bírák bármely valódi részcsoportja semmit se tudjon mondani a választók szavazatairól.

6.7.4. Bizonyíthatóság

Bárki (akár kívülálló is) bármikor tudja bizonyítani, hogy egy adott választó helyes szavazatot adott e le, és ezt a bizonyítást elvégezve csak annyi információhoz jusson, hogy a szavazás helyes volt e, semmit ne tudjon meg arról, hogy az adott választó mit választott.

6.8. Szavazási séma

Olyan folyamatleírás, amely tartalmazza a szavazás menetét a követelményeknek megfelelően.

7. fejezet

Két-kimenetű szavazási séma[2]

Ebben a részben egy olyan szavazási sémát tárgyalunk, melyben a szavazók két lehetőség közül döntik el, hogy melyiket választják, majd ezeket átadják a bizottság tagjainak oly módon, hogy azok ne tudják meg, hogy egy adott szavazó mit választott.

7.1. Kezdőállapot

Legyen E elliptikus görbe Z_p felett. Legyenek $g, G \in E$ az E elliptikus görbe generátorelemei. A bizottság t bizottsági tagból áll, minden egyes tagjának van egy nyilvános kulcsa (a görbe egy-egy pontja): $h_j = [z_j]g$, ahol z_j titkos kulcsa a j. bizottság-tagnak.

7.2. Szavazás

Az i. szavazó szavazata legyen: $v_i \in \{0, 1\}$ és választ mellé egy $s_i \in Z_p$ random értéket, és nyilvánosságra hozza az $U_i = [s_i + v_i]g$ pontot. A titokmegosztás módszerével a $[s_i]g$ pontot megosztja a bizottság tagjai közt:

Titokmegosztás módszere: Minden egyes szavazó választ egy Z_p felett értelmezett $t - 1$-edfokú random polinomot:

$$p_i(x) = \sum_{k=0}^{t-1} \alpha_{i,k} x^i$$

ahol $\alpha_{i,0} = s_i$ és $\alpha_{i,1}, \ldots, \alpha_{i,t-1} \in Z_p$.

A polinomot csak ő tudja, senki más. Nyilvánosságra hozza a $C_{i,k} = [\alpha_{i,k}]G$, $0 \leq k \leq t-1$ pontokat, valamint megosztja a $H_{i,j} = [p_i(j)]h_j$ pontokat. Ezen felül megmutatja, hogy a megosztott titok konzisztens, kiszámolva a

$$X_{i,j} = \sum_{k=0}^{t} [j^k] C_{i,k} = [\sum_{k=0}^{t-1} \alpha_{i,k} j^k] G$$

pontot, átlátszó bizonyítással igazolja, hogy $\log_G X_{i,j} = \log_{h_j} H_{i,j}$.
Megmutatja, hogy $v_i \in \{0, 1\}$ szintén átlátszó bizonyítással bizonyítva a

$$log_G C_0 = log_g U_i \bigvee log_G (G + C_0) = log_g U_i$$

$$C_0 = [s_i]G$$

logikai kifejezést. Ezzel azt akarjuk bizonyítani, hogy a szavazó helyes szavazatot adott-e le. Itt a vizsgáló személy bárki lehet, a bizonyító az a szavazó, akinek a szavazatát vizsgálják.

7.3. Szavazatok összeszámlálása

Feltéve, hogy a szavazók helyesen szavaztak (ezt minden esetben tudták bizonyítani), az összeszámlálás a következőképpen történik: Minden egyes bizottságbeli tag kiszámolja a következő értéket:

$$H_j^* = \sum_i H_{i,j} = \sum_i [p_i(j)] h_j = [\sum_i p_i(j)] h_j.$$

Ezt dekódolják a következőképpen:

$$S_j^* = [1/z_j] H_j^* = [\sum_{i=1}^{n} p_i(j)] g.$$

Ezek még csak a megosztott rész-titkok, maguk a szavazatok Lagrange interpolációval megkaphatók:

$$\sum_{j \in A} [\lambda_{j,A}] S_j^* = \sum_{j \in A} [\lambda_{j,A}] (\sum_i [p_i(j)] g)$$

$$= [\sum_{j \in A} \sum_i p_i(j) \lambda_{j,A}] g$$

$$= [\sum_i p_i(0)]g = [\sum_i s_i]g,$$

ahol A a bizottság tagjainak halmaza, $\lambda_{j,A}$ pedig

$$\lambda_{j,A} = \sum_{l \in A - \{j\}} \frac{l}{l-j}.$$

Ha már megvan $[\sum_i s_i]g$, kiszámolva a $\sum_i U_i$ - t, hozzáadva ehhez $[\sum_i s_i]g$ pont inverzét kapjuk a $[\sum_i v_i]g = [T]g$ - t, ahonnan $T = \sum_i v_i$ kiszámolható a 3.3. fejezetben tárgyalt algoritmusok egyikével, mivel T viszonylag kicsi érték (ha a világon minden ember szavaz, akkor is maximum 2^{33} szavazat érkezik).

8. fejezet

Több-kimenetű szavazási séma

Ebben a részben tárgyalt szavazási sémában már több lehetőség közül dönthet a választó.

8.1. Kezdőállapot

Legyen E elliptikus görbe Z_p felett. Legyenek $g, G \in E$ az E elliptikus görbe generátorelemei. A bizottság t bizottsági tagból áll, minden egyes tagjának van egy nyilvános kulcsa (a görbe egy-egy pontja): $h_j = [z_j]g$, ahol z_j titkos kulcsa a j. bizottság-tagnak. Több-kimenetű szavazás esetén abban az esetben, ha mindenki csak egyet választhat a szavazati jelöltek közül, minden egyes jelölthöz létre kell hozni egy $0-1$-ekből álló $y-1$ dimenziós vektort, ahol y a jelöltek száma. Pl.:

$$J_0 := (0, 0, 0, 0, \ldots, 0),$$

$$J_1 := (1, 0, 0, 0, \ldots, 0),$$

$$J_2 := (0, 1, 0, 0, \ldots, 0),$$

$$J_3 := (0, 0, 1, 0, \ldots, 0),$$

$$\ldots$$

$$J_{y-1} := (0, 0, 0, 0, \ldots, 1).$$

8.2. Szavazás

Az i. szavazó szavazatához rendelt vektor l. tagja legyen: $v_{i,l} \in \{0,1\}$ és választ mellé egy $s_{i,l} \in Z_p$ random értéket, és nyilvánosságra hozza az $U_{i,l} = [s_{i,l} + v_{i,l}]g$ pontokat. A titokmegosztás módszerével a $[s_{i,l}]g$ pontokat megosztja a bizottság tagjai között:

Titokmegosztás módszere: Minden egyes szavazó választ $y-1$ Z_p felett értelmezett $t-1$-edfokú random polinomot:

$$p_{i,l}(x) = \sum_{k=0}^{t-1} \alpha_{i,k,l} x^i,$$

ahol $\alpha_{i,0,l} = s_{i,l}$ és $\alpha_{i,1,l}, \ldots, \alpha_{i,t-1,l} \in Z_p, l = 1 \ldots y-1$.

A polinomokat csak ő tudja, senki más. Nyilvánosságra hozza a $C_{i,k,l} = [\alpha_{i,k,l}]G$, $0 \leq k \leq t-1$ pontokat, valamint megosztja a $H_{i,j,l} = [p_{i,l}(j)]h_j$ pontokat. Ezen felül megmutatja, hogy a megosztott titok konzisztens, kiszámolva a

$$X_{i,j,l} = \sum_{k=0}^{t-1} [j^k] C_{i,k,l} = [\sum_{k=0}^{t-1} \alpha_{i,k,l} j^k] G, l = 1 \ldots y-1$$

pontot, átlátszó bizonyítással igazolja, hogy $\log_G X_{i,j,l} = \log_{h_j} H_{i,j,l}$.

Megmutatja, hogy $v_{i,l} \in \{0,1\}$ szintén átlátszó bizonyítással bizonyítva a

$$\log_G C_{0,l} = \log_g U_i \bigvee \log_G (G + C_{0,l}) = \log_g U_i$$
$$C_{0,l} = [s_{i,l}]G, l = 1 \ldots y-1$$

logikai kifejezést. Ezen kívül ad egy átlátszó bizonyítást a

$$\bigwedge_{q=1}^{y-1} (\log_G C_{0,q} = \log_g U_i \bigvee \log_G (G + C_{0,q}) = \log_g U_i)$$

logikai kifejezésre.

8.3. Szavazatok összeszámlálása

Feltéve, hogy a szavazók helyesen szavaztak (ezt minden esetben tudták bizonyítani), az összeszámlálás a következőképpen történik: Minden egyes bizottságbeli tag kiszámolja a következő értéket:

$$H^*_{j,l} = \sum_i H_{i,j,l} = \sum_i [p_{i,l}(j)] h_j = [\sum_i p_{i,l}(j)] h_j.$$

Ezt dekódolják a következőképpen:

$$S^*_{j,l} = [1/z_j]H^*_{j,l} = [\sum_{i=1}^{n} p_{i,l}(j)]g, l = 1 \dots y-1.$$

Ezek még csak a megosztott rész-titkok, maguk a szavazatok Lagrange interpolációval megkaphatók:

$$\sum_{j\in A}[\lambda_{j,A}]S^*_{j,l} = \sum_{j\in A}[\lambda_{j,A}](\sum_i [p_{i,l}(j)]g)$$

$$= [\sum_{j\in A}\sum_i p_{i,l}(j)\lambda_{j,A}]g$$

$$= [\sum_i p_{i,l}(0)]g = [\sum_i s_{i,l}]g,$$

ahol A a bizottság tagjainak halmaza, $\lambda_{j,A}$ pedig

$$\lambda_{j,A} = \sum_{l\in A-\{j\}} \frac{l}{l-j}.$$

Ha már megvan $[\sum_i s_{i,l}]g$, kiszámolva a $\sum_i U_{i,l}$ - t, hozzáadva ehhez $[\sum_i s_{i,l}]g$ pont inverzét kapjuk a $[\sum_i v_{i,l}]g = [T_l]g$ - t, ahonnan $T_l = \sum_i v_{i,l}, l = 1 \dots y-1$ kiszámolható a 3.3. fejezetben tárgyalt algoritmusok egyikével, mivel a T_l-k viszonylag kicsi értékek (ha a világon minden ember szavaz, akkor is maximum 2^{33} szavazat érkezik).
$T_1, T_2, \dots, T_{y-1}$ értékeket kapunk,amikből megkaphatók a jelöltekre leadott szavazatok számai:

$$\sum J_0 = n - T_1 - T_2 - T_3 - \dots - T_{y-1}$$

$$\sum J_1 = T_1$$

$$\sum J_2 = T_2$$

$$\sum J_3 = T_3$$

$$\dots$$

$$\sum J_{y-1} = T_{y-1}.$$

Ez egy egyenletrendszer, melynek megoldásai lesznek az egyes jelöltekre leadott szavazatok számai.

9. fejezet

Általános szavazási séma

Ebben a részben egy olyan választási sémát nézünk meg, amely bármilyen követelményt képes kielégíteni.
Lehetséges, hogy több témában kell döntést hozni egy választás alkalmával, minden témában lehet több választási lehetőség, és akár az is lehetséges, hogy egy témán belül több választ is megengedünk. Ezeket az igényeket kontrollálni kell, bizonyítást kell adni arra, hogy egy adott választó ezen kritériumoknak megfelelően szavazott, érvényes szavazatot adott le.

9.1. Kezdőállapot

Legyen E elliptikus görbe Z_p felett. Legyenek $g, G \in E$ az E elliptikus görbe generátorelemei. A bizottság t bizottsági tagból áll, minden egyes tagjának van egy nyilvános kulcsa (a görbe egy-egy pontja): $h_j = [z_j]g$, ahol z_j titkos kulcsa a j. bizottság-tagnak. Többértékű szavazás esetén a választó szavazata, miután a szavazás megtörtént, egy y dimenziós J vektor lesz, ahol minden egyes érték egy $0, 1$ halmazbeli elem, minden ilyen érték azt mutatja, hogy egy adott témában egy adott választ a választó igennek, vagy nemnek jelölt e. Pl.:

$$J := (1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, \ldots, 0, 1).$$

Legyen Φ a J bitjeinek egy logikai kifejezése, amely igaz, ha a szavazó helyesen szavazott, hamis, ha nem szavazott helyesen.

9.2. Szavazás

Az i. szavazó szavazatához rendelt vektor l. tagja legyen: $v_{i,l} \in \{0,1\}$ és választ mellé egy $s_{i,l} \in Z_p$ random értéket, és nyilvánosságra hozza az $U_{i,l} = [s_{i,l} + v_{i,l}]g$ pontokat. A titokmegosztás módszerével a $[s_{i,l}]g$ pontokat megosztja a bizottság tagjai között:

Titokmegosztás módszere: Minden egyes szavazó választ y darab Z_p felett értelmezett $t-1$-edfokú random polinomot:

$$p_{i,l}(x) = \sum_{k=0}^{t-1} \alpha_{i,k,l} x^i,$$

ahol $\alpha_{i,0,l} = s_{i,l}$ és $\alpha_{i,1,l}, \ldots, \alpha_{i,t-1,l} \in Z_p, l = 1 \ldots y$.

A polinomokat csak ő tudja, senki más. Nyilvánosságra hozza a $C_{i,k,l} = [\alpha_{i,k,l}]G$, $0 \leq k \leq t-1$ pontokat, valamint megosztja a $H_{i,j,l} = [p_{i,l}(j)]h_j$ pontokat. Ezen felül megmutatja, hogy a megosztott titok konzisztens, kiszámolva a

$$X_{i,j,l} = \sum_{k=0}^{t-1} [j^k] C_{i,k,l} = [\sum_{k=0}^{t-1} \alpha_{i,k,l} j^k] G, l = 1 \ldots y$$

pontot, átlátszó bizonyítással igazolja, hogy $\log_G X_{i,j,l} = \log_{h_j} H_{i,j,l}$.

Megmutatja, hogy $v_{i,l} \in \{0,1\}$ szintén átlátszó bizonyítással bizonyítva a

$$\log_G C_{0,l} = \log_g U_i \bigvee \log_G (G + C_{0,l}) = \log_g U_i$$

$$C_{0,l} = [s_{i,l}]G, l = 1 \ldots y$$

logikai kifejezést.
Ezen felül ad egy átlátszó bizonyítást Φ logikai kifejezésre, melyben a J vektor bitjei helyett az ezekhez tartozó diszkrét elliptikus logaritmikus kifejezés szerepel.

9.3. Szavazatok összeszámlálása

Feltéve, hogy a szavazók helyesen szavaztak (ezt minden esetben tudták bizonyítani), az összeszámlálás a következőképpen történik: Minden egyes bizottságbeli tag kiszámolja a következő értéket:

$$H^*_{j,l} = \sum_i H_{i,j,l} = \sum_i [p_{i,l}(j)] h_j = [\sum_i p_{i,l}(j)] h_j.$$

Ezt dekódolják a következőképpen:

$$S^*_{j,l} = [1/z_j]H^*_{j,l} = [\sum_{i=1}^{n} p_{i,l}(j)]g, l = 1 \dots y.$$

Ezek még csak a megosztott rész-titkok, maguk a szavazatok Lagrange interpolációval megkaphatók:

$$\sum_{j\in A}[\lambda_{j,A}]S^*_{j,l} = \sum_{j\in A}[\lambda_{j,A}](\sum_i [p_{i,l}(j)]g)$$

$$= [\sum_{j\in A}\sum_i p_{i,l}(j)\lambda_{j,A}]g$$

$$= [\sum_i p_{i,l}(0)]g = [\sum_i s_{i,l}]g,$$

ahol A a bizottság tagjainak halmaza, $\lambda_{j,A}$ pedig

$$\lambda_{j,A} = \sum_{l\in A-\{j\}} \frac{l}{l-j}.$$

Ha már megvan $[\sum_i s_{i,l}]g$, kiszámolva a $\sum_i U_{i,l}$ - t, hozzáadva ehhez $[\sum_i s_{i,l}]g$ pont inverzét kapjuk a $[\sum_i v_{i,l}]g = [T_l]g$ - t, ahonnan $T_l = \sum_i v_{i,l}, l = 1 \dots y-1$ kiszámolható a 3.3. fejezetben tárgyalt algoritmusok egyikével, mivel a T_l-k viszonylag kicsi értékek (ha a világon minden ember szavaz, akkor is maximum 2^{33} szavazat érkezik).
$T_1, T_2, \dots, T_y$ értékeket kapunk, ahol T_i a szavazatok i. bitjeinek összegei, azaz az ezt a választási lehetőséget igennel megjelölők száma. Ebből a nem szavazatok száma $n - T_i$ kifejezésekkel megkaphatók.

10. fejezet

Implementáció

Ezen szavazási eljárás programozása és tesztelése során egy 2410 GHz órajelű, 768 MB RAM-mal rendelkező számítógépet használtam, Microsoft Windows XP operációs rendszerrel, Microsoft Visual Studio.NET programozási környezetben.
Ma a biztonságosnak tartott kulcs méret 160 bit, amin a program dolgozik, egy 512 bites prím rendű test feletti elliptikus görbe, tehát a kulcs méret 512 bites. A generátor pontból egy tetszőleges pont kiszámítása átlagosan 7 másodpercet vett igénybe. Mind az elliptikus görbét, mind a hozzá tartozó generátor elemet a program generálta.
A program leírása:

A szerver működése:
A felhasználókat három szerepkörbe sorolja:
- *Adminisztrátor:* Egyetlen ilyen felhasználó van, aki létrehozni, szerkeszteni, törölni tud szavazásokat, felhasználókat.
- *Bizottsági tag:* A szavazási eljárás során azon felhasználók szerepköre, akik a szavazás tisztaságáért felelősek. Ők nem szavazhatnak.
- *Szavazó:* A legtöbb felhasználó szerepköre, azon személyek, akik a szavazás során lehetőséget kaphatnak a szavazásra. Ha valaki regisztrálja magát a szerveren, automatikusan ebbe a szerepkörbe kerül.

Több szavazást is tud kezelni egyszerre, indítás után betölti a konfigurációs állományból a még el nem kezdődött, a már befejeződött és az éppen folyó szavazásokat.
A még el nem kezdődött szavazásokra bizottsági tag szerepkörrel rendelkező felhasználók kijelölését az adminisztrátor végzi.

Az aktuálisan futó szavazások esetén olyan szavazó szerepkörrel rendelkező felhasználók szavazatait várja, akik jogosultak az adott szavazáson való részvételre.
A befejeződött szavazások végeredménye publikus, bármely szerepkörrel rendelkező felhasználó láthatja. a konfigurációs fájlon kívül található még egy, a titkosításhoz használt elliptikus görbét tartalmazó fájl, valamint minden egyes felhasználóhoz található egy fájl, ami a publikus kulcsát tartalmazza.

A kliens működése:
Indításkor végignézi a kliens számítógépen található meghajtókat, és megkeresi az azonosításhoz szükséges fájlt, amely a felhasználó privát kulcsát tartalmazza. Azonosításkor a szerver egy nonce-ot (number used once) generál, amit a kliensnek alá kell írnia a privát kulcsával. Ha a szerver a kapott értéket a publikus kulccsal visszafejtve visszakapja a nonce-ot, akkor beengedi a felhasználót. A felhasználó a saját szerepköréhez tartozó lehetőségeket tudja csak végrehajtani, és egy szavazó egy szavazásokra csak akkor tud szavazni, ha a szavazás éppen aktuális, a felhasználó jogosult a szavazásra, és még nem szavazott rajta.

A titkosító elliptikus görbe:

$Y^2 = X^3 + 6220401646950252931576064500844786003486366754404096318977926714883970299729218781389947437095177723132645102940728802504999299221432237085314049345650446X + 6463257647766091213259986053340374603834997968381094928302672964370089052812323845482279436914634898145692360012715258246401278605381381185898512313940447$

modulo:

$8922549574716725653246538567936964536491753564714409475378788485310300876701119237136101075205971851238397487625457232844279788733922205434632380099202349$

generátorelem:

$(1728735741224507109031947617646445771597198913570780296321767918521765509737704126825100107932354081812463256842011097160077927142702985142168399396683653, 4715297012050682908596748304267276366337209549410698904591087389629394614962887254193472649287582405576356717313545859993148524646884110648860296322345259)$

11. fejezet

Függelék - Történeti áttekintés[7]

Most tekintsük át, milyen változásokon ment keresztül a kriptográfia az emberiség történelme során. A teljesség igénye nélkül felsoroljuk a fontosabb eseményeket, melyeknek köszönhetően a kriptográfia eljutott mai arculatához. Segítségével átláthatjuk ezen dolgozat előzményeit, célkitűzéseit.

11.1. Ókor

ie. 3500 körül A sumérok elkezdték használni az ékírást, az egyiptomiaknál pedig kialakultak a hieroglifák.

ie. 1500 körül A főníciaiak létrehozták az ábécét.

ie. 600-500 Héber tudósok egyszerű monoalfabetikus rejtjeleket szerkesztettek (például az Atbash).

ie. 400 körül Hérodotosz feljegyezte, hogy egy palatáblán lévő üzenetet viasszal lefedve sikerült eljuttatni Perzsiából Görögországba (szteganográfia). Lüszandrosz spártai hadvezér a szkütalé szalag révén kapott információkat ie. 404-ben a perzsák várható támadásáról. Szkütalé módszer az első ismert katonai rejtjelező módszer. Egy szíjat vagy szalagot egy szabályos sokszögalapú hasábra, pálcára csavarták fel, majd erre a pálca tengelyének irányában felírták az szöveget. Letekerve a szíj csak betűk értelmetlen sorozata mindaddig, míg a címzett fel nem tekeri egy ugyanolyan átmérőjű rúdra.

ie. 50 körül Római rejtjelek, mint például a Caesar-rejtjel.

11.2. Középkor

800-as évek	A Korán szövegének tanulmányozása közben arab tudósok a gyakoriságelemzés módszerét kifejlesztve megfejtettek monoalfabetikus helyettesítéssel rejtjelezett üzeneteket.
1450-1520	Létrejött a Voynich kézirat, egy ismeretlen betűkkel és nyelven írt könyv, melynek tartalmát mind a mai napig nem sikerült megfejteni.[5]
1466	Leon Battista Alberti elkészítette az első általunk is ismert polialfabetikus rejtjelet, melyhez egy dekóder-készüléket is feltalált.
1499	Johannes Trithemius megírta a kriptográfiával és szteganográfiával foglalkozó híres könyvét.
1553	Giovan Batista Belaso a La cifra del. Sig. Giovan Batista Belaso című könyvében leírta a később Vigene‘re-rejtjelnek elnevezett módszert.
1585	Blaise de Vigene‘re újra felfedezte és közzétette a Belaso féle rejtjelezés egy kicsit erősebb változatát.
1586	Kriptoanalízis segítségével bizonyította rá a Babington-összeesküvés résztvevőire bűnösségüket Sir Francis Walsingh, I. Erzsébet angol királynő államminisztere.
1645 körül	Megjelent John Wilkins Mercury című angol nyelvű könyve a kriptográfiáról.

11.3. Újkor

1793	Claude Chappe létrehozta az első nagytávolságú szemafor jelzéseket használó kor
1795	Thomas Jefferson megalkotta a Jefferson korongok rejtjelező szerkezetet és módszert, mely azonban csak akkor vált ismertté Bazerie Cilinderek néven, amikor egy évszázad múlva Etienne Bazeries újra felfedezte.
1809-14	A Félszigeti háború alatt George Scovell, Wellington vezérkarának egyik tisztje kapta a feladatot, hogy megfejtse a franciák rejtjelezett üzeneteit.
1832	Létrehozták az elektromágneses telegráfot.
1837	Samuel Morse megtervezte és szabadalmaztatta az elektromos telegráfot, megalkotta a Morze kódot.
1854	Charles Wheatstone feltalálta a digrafikus Playfair-rejtjelet. Charles Babbage megfejtette az addig feltörhetetlennek tartott Vigene're-rejtjelet.
1883	Auguste Kerckhoffs megírta La Cryptographie militare (A katonai titkosírás) című tanulmányát.
1885	Nyilvánosságra kerültek a Beale-papírok, melyek kriptográfusok és kincsvadászok nemzedékeit késztette fejtörésre.
1890-es évek	Többen egymástól függetlenül feltalálták a szikratávírót, vagy ahogy később elnevezték, a rádiót.

11.4. A XX. század első fele

A világháborúk, majd a hidegháború valósággal gondolatháborút indított el a harcoló felek titkosítással foglalkozó szakemberei között, amely óriási lendületet adott a matematika és az informatika fejlődésének.

1915 körül	William Friedman a matematikai statisztika módszereit használta fel a kriptoanalízisben (kappa-teszt stb.).
1917	Gilbert Vernam feltalálta a távgépíró elvén alapuló, a kulcsot lyukasztott távírószalag formájában tároló rejtjelező gépet, majd Joseph Mauborgne-nel együtt az egyszeri kulcsos módszert. Az angol rejtjelfejtők dekódolták a német külügyminiszternek, Zimmermann-nak a mexikói elnökhöz intézett táviratát, amelynek következtében az Amerikai Egyesült Államok belépett az I. világháborúba.
1919	Arthur Scherbius feltalálta és szabadalmaztatta az első forgó keverőtárcsákra épülő rejtjelező gépet, melyet később Enigmának neveztek el. Vele gyakorlatilag egyidőben három másik feltaláló is hasonló gépet épített: Alexander Koch, Arvid Damm és Edward Hebern.
1931	Megjelent Herbert O. Yardley Az amerikai fekete szoba című könyve, ami az általa vezetett, 1913 és 1929 között működő szervezet (MI-8) tevékenységéről szólt.
1932	A lengyel Marian Rejewski feltöri a német hadsereg Enigmával kódolt üzeneteit.
1940	Az amerikai hadsereg kódfejtő részlege (SIS) feltörte a japánok Purple elnevezésű gépének a kódját. Alan Turing megtervezte az elektromechanikus Turing-bombát, melyekkel az Állami Rejtjelező és Rejtjelfejtő Iskola bombakezelői néhány óra alatt képesek voltak az Enigma naponta változó alapbeállításait és kódjait feltörni.
1941 december	A JN-25-ös japán kód hirtelen megváltoztatása révén a japán csendes-óceáni flotta meglepte és elpusztította a Pearl Harborban állomásozó amerika hadihajókat, az USA belépett a II. világháborúba.

1942 Az amerikai hadsereg navahó indiánokat képezett ki rádiósnak, akik anyanyelvükön továbbították a kódszavakkal tarkított üzeneteket.
Az amerikaiak megfejtették a JN-25 új verzióját, ami jelentős szerepet játszott a fordulópontot jelentő Midway-i csata megnyerésében.

1943 Max Newman, Wynn-Williams, és a GSCS-beli csapatuk befejezte a Heath Robinson elnevezésű speciális kódtörő gépet.
A Bletchley Parkban dolgozó Thomas Flowers a német Lorenz kód (SZ42) feltörésére létrehozta a Colossus nevű gépet, amely tekinthető a legelső programozható számítógépnek.

1946 A Venona projekt első sikeres betörése a magasszintű szovjet diplomáciai hírszerzés adatforgalmába.

1948 Megjelent Claude Shannon A kommunikáció matematikai elmélete (Mathematical Theory of Communication) című munkája, mely az információelmélet legalapvetőbb törvényeit tartalmazta.

11.5. A XX. század második fele

Megjelentek a nagy teljesítményű számítógépek, ami további lökést adott a kriptográfiának, mivel segítségükkel bonyolult és a hagyományos módszerekkel fejthetetlennek tűnő kódokat lehetett feltörni és előállítani.
A XX. század második felében a kriptográfia tudománya szorosan összefonódott a számítástechnika fejlődésével, az internet- és mobilkommunikáció által alkalmazott adattitkosítási algoritmusokkal.

1952 Megalapították az Amerikai Egyesült Államok kormányának kriptológiai szervezetét, a Nemzetbiztonsági Ügynökséget (NSA).

1957 Az NSA létrehozta a TSEC/KW-26, ROMULUS kódnevű titkosítási rendszert, melyet az USA, majd később a NATO országok is használtak. (Ez váltotta fel az olyan régebbi forgótárcsás vagy elektromechanikus rendszereket, mint a SIGABA és az angol 5-UCO.)

1964 Megjelent David Kahn Kódfeltörők (The Codebreakers) című könyve, amely először foglalkozott a kódok fejlődésével.

1968 John Anthony Walker információkat adott el a Szovjetuniónak a KL-7, ADONIS kódnevű forgótárcsás kódgépről. (Walker csak 1985-ben bukott le, a KL-7-et azután nem használták.)

1969 Rákapcsolták az internet elődjének számító, IP-alapú ARPANET hálózatra az első szervert.

1974 Horst Feistel, az IBM kriptográfusa létrehozta a Feistel féle általános hálózati blokk kódoló eljárást.

1976 Az IBM által kifejlesztett és publikált DES (Data Encryption Standard) lett az Egyesült Államok hivatalos Szövetségi Információs Szabványa. Megjelent Whitfield Diffie és Martin Hellman Új irányzatok a kriptográfiában (New Directions in Cryptography) című könyve, mely egy radikálisan új, az úgynevezett kulcs cserén alapuló kriptográfia fogalmát vezette be.

1977 Ron Rivest, Adi Shamir és Len Adleman kifejlesztették az RSA eljárást.

1981 Richard Feynman elméletben megtervezte a kvantum számítógépet.

1987 Neal Koblitz javaslatot tett az elliptikus görbék kriptográfiában való alkalmazására.[3]

1986 A kormányzati és vállalati kompjúterekkel szembeni egyre sokasodó támadások és betörési kísérletek után az Egyesült Államok Kongresszusa elfogadta a Computer Fraud and Abuse Act (Számítógéppel elkövetett csalás és visszaélés) törvényt, amely bűncselekménynek nyilvánította a számítógépes rendszerekbe való illetéktelen behatolást.

1988	Kifejlesztették az első optikai chipet.
1989	Tim Berners-Lee és Robert Cailliau a CERN-ben létrehozta a későbbi Világháló prototípusát.
1991	Phil R. Zimmermann nyilvánossá tette az általa kifejlesztett PGP nevű publikus kulcsú titkosítási programot, amely a legelterjedtebb e-mail titkosító szoftver lett a világon. Az RSA Laboratórium elindította az RSA Faktorizációs Versenyt, hogy bátorítsa a kutatást a számítógépes számelmélet és a nagy számok faktorizációjának gyakorlati nehézségei terén.
1994	Megjelent Bruce Schneier Alkalmazott kriptográfia (Applied Cryptography) című műve. A Netscape kibocsátotta a Secure Sockets Layer (SSL) titkosító protokolját. Peter Shor kigondolt egy algoritmust, amely alapján a kvantum számítógépek képesek meghatározni nagy számok faktorizációját. (Ez volt az első olyan érdekes probléma, ahol a kvantum számítógépek jelentős sebesség növekedéssel kecsegtettek, és ez nagyban növelte az iránt uk való érdeklődést, főleg a kriptográfusok körében.) A régebben védett, de nem szabadalmaztatott RC4 titkosító algoritmust közzétették az interneten.
1995	Az NSA kiadta a SHA1 hash algoritmust, mint az általuk kifejlesztett Digitális Aláírás Szabvány (Digital Signature Standard) részét.
1997	Kiadták az OpenPGP specifikációját. Az Usenet közzétette a Ciphersaber nevű szimmetrikus kulcsú titkosító eljárást, mely olyan egyszerű volt, hogy algoritmusát emlékezetből rekonstruálni lehetett.
1999 október	Megjelent a DeCSS számítógépes program, mely képes volt feltörni a CSS (Content-Scrambling System) kódolású video DVD tartalmakat.

11.6. XXI. század

2000 Az Egyesült Államok kormánya enyhített a kriptográfia exportjára vonatkozó korlátozásán.
Az RSA Security Inc. néhány nappal a szabadalom lejárta előtt publikussá tette az általuk használt RSA algoritmust. (Ez, az export tilalom enyhítésével együtt ledöntötte az utolsó korlátot is a szoftverek világhálós disztribúciója előtt.)
Az angol nyomozói erők szabályozásáról szóló törvény (Regulation of Investigatory Powers Act) kötelez mindenkit arra, hogy hivatalos kérés esetén kiszolgáltassa kriptográfiai kulcsait az arra jogosult személynek.
Elindult az európai NESSIE (New European Schemes for Signatures, Integrity and Encryption) és a japán CRYPTREC elnevezésű projekt, melynek célja a biztonságos titkosítás alapjainak meghatározása volt. (Mindkettő 2003-ban fejeződött be.)

2001 Az amerikai Nemzeti Szabványügyi és Technológiai Intézet (National Institute for Standards and Technology) öt évig tartó elemzés után a belga Rijndael algoritmust választotta az Egyesült Államok új hivatalos titkosítási szabványául, neve AES (Advanced Encryption Standard).

2002 10000 számítógéppel 540 nap alatt sikerült feltörni egy 109 bites prím test feletti elliptikus görbén alapuló titkosítási sémát.[8]

2004 2600 számítógéppel 17 hónap alatt sikerült feltörni egy 109 bites kettő hatvány rendű test feletti elliptikus görbén alapuló titkosítási sémát.[9]

2005 amerikai FBI ügynökök bebizonyították, hogy képesek feltörni a WEP (Wired Equivalent Privacy), rádiós hálózatok titkosítási eljárását mindenki által elérhető eszközök segítségével.

2007 Szabadon letölthetővé válik egy, a szivárványtábla-módszert alkalmazó szoftver, amely az operációs rendszerből kibányászott és egy adatbázissal összevetett, eddig biztonságosnak hitt jelszavakat másodpercek alatt feltöri.

2008 Ma az RSA-nak, a legelterjedtebb aszimmetrikus titkosítási sémának a biztonságos kulcshossza 1024 bit, míg az ECC esetében ez a méret csak 160 bit.

Irodalomjegyzék

[1] Susanne Schmitt, Horst G. Zimmer: *Elliptic Curves - A Computational Approach*, de Gruyter Studies in Mathematics, (2003)

[2] Zuzana Rjasková: *Electronic Voting Shemes*, Department of Computer Science Faculty of Mathematics, Physics and Informatics - Comenius University, Bratislava (2002)

[3] Neal Koblitz, Elliptic curves cryptosystems, Math. Comp. 48(177): 203-209 (1987)

[4] Nagy P., Nyilvános kulcsú titkosítások (Public Key Cryptosystems). Diploma Work, University of Debrecen (2000)

[5] Voynich, Wilfrid Michael: *A Preliminary Sketch of the History of the Roger Bacon Cipher Manuscript*, Transactions of the College of Physicians of Philadelphia 3(43): 415-430 (1921)

[6] Johannes Trithemius: *Steganographie: Ars per occultam Scripturam animi sui voluntatem absentibus aperiendi certu* (Written: 1500. First printed edition: Frankfurt, 1606)

[7] Wikipédia - Kriptográfia, A kriptográfia története -

`http://hu.wikipedia.org/wiki/Kriptogr%C3%A1fia`

[8] Certicom Announces Elliptic Curve Cryptosystem Challenge Winner for ECC over prime field -

`http://www.certicom.com/index.php?action=company,press_archive&view=121`

[9] Certicom Announces Elliptic Curve Cryptography Challenge Winner for ECC over binary field -

`http://www.certicom.com/index.php?action=company,press_archive&view=307`